MSX-DOS スーパーハンドブック

BITS著

アスキー出版局

目　次

第1章　MSX-DOSの特徴を知る

第2章　MSX-DOSを便利に使う

第3章　MSX-DOS用開発ツール

第4章　MSX-DOSのコマンドを作る

第5章　MSX-DOSなんでも質問箱

第6章　MSX-DOSユーティリティー集

プログラムリスト

第3章

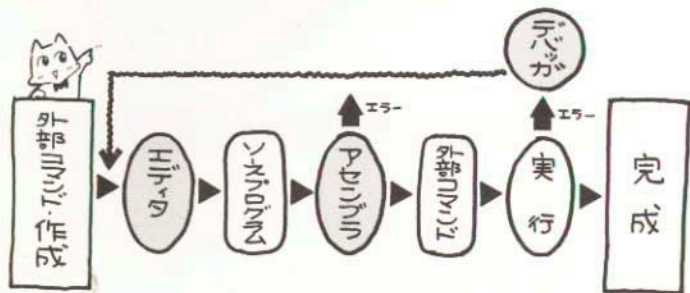
MSX-DOS用 開発ツール

3.1 MSX-DOSで開発するには

●開発ツールをそろえる

DOSでは、いろいろな言語を使用することができます。そして、その目的もさまざまです。例えば、ディスクBASIC用のマシン語プログラムを開発することも考えられますし、LISPなどといった高級言語を使用して人工知能分野に関わるプログラムを作ることもできます。また、クロスアセンブラなどを使えば、他CPUのマシン語開発などもできます。そして、実際にプログラム開発を行おうとすると、目的に応じて必要となるツール類も当然のことながら変わってきます。そこで、ここで行う目的と必要となるツールを示しておきましょう。

まず目的ですが、ここでは「MSX-DOSで動作する外部コマンドを作る」ことに限らさせてもらいます。となると、ツール類も決まってきます。外部コマンドは基本的にマシン語プログラムなので、これを生成するようなツールが必要となるのです。これにはCコンパイラといった高級言語を使用することも考えられますが、ここではアセンブラでの開発環境を提供していきます。

アセンブラを使用して開発するために必要なツールは、ソースプログラムを作るための「テキストエディタ」、マシン語に変換するための「アセンブラ」、そして動作をチェックするための「デバッガ」があれば事足ります。これらは市販品(M80、MED、S-BUGなど)を

使用しても構いませんが、手に入らない人もいることを想定して、本書がオリジナル開発したものを掲載することにします。むろん、これらのツールは外部コマンドでありDOS上でのみ利用可能です。それでは、それらの入力の仕方、使い方を解説していきます。なお、市販品とは使用法が異なりますので注意してください。

●プログラム入力のしかた

入力はP154掲載の「COMファイル入力用ツール」を使用して行います。プログラムおよび操作説明はそちらを見てください。

掲載した「アセンブラ」、「エディタ」、「デバッガ」の入力するアドレスと、セーブするときに指定するファイル名、および掲載ページを以下に示します。間違えないようにしてください。

・アセンブラ	ASM.COM	0100H～1B8FH	P161
・エディタ	PED.COM	0100H～277FH	P168
・デバッガ	BUG.COM	0100H～23FFH	P178

3.2 MSX-DOS版アセンブラ

●アセンブラの特徴

ASM.COM

書式 ASM_<ファイル名>_ [スイッチ1] [スイッチ2]

Z80CPU用の2パスアセンブラで、アセンブリ記述されたプログラムをマシン語に変換します。本アセンブラは「ポケットバンクシリーズ　マシン語入門」に掲載した「モニタアセンブラ」のアセンブラの部分を外部コマンド化し、さらに使いやすく改良したものです。主な特徴は次のとおりです。

・アセンブル作業が簡単である

本アセンブラはORG命令を0100hとしておくことによりアセンブル後自動的にCOMファイルが生成され、すぐに外部コマンドができるようになっています。そのため、市販のアセンブラのようなリンク作業がいらないなど、シンプルに作業が運べます。

・モニタアセンブラのソースがそのまま入力として使える

モニタアセンブラで作成した行番号付きのソースプログラムを、そのままアセンブルすることができます。これにより、ディスクBASIC上で開発したプログラムをすぐにMSX-DOSで利用することができます。

・BSAVE形式のファイルが出力できる

アセンブルして、すぐにディスクBASICに移行し、BLOAD命令により読み込むといったことが、オプションスイッチによりできるようになっています。この機能を使えば、簡単にディスクBASICで動作するマシン語が開発できます。

・ファイルに出力されるオブジェクト

「モニタアセンブラ」では、高速かつ、操作性のよい環境をディスクBASIC上で実現するため、ソースプログラム、オブジェクトプログラムをオンメモリ上ですべて処理していましたが、反面メモリ不足で大きなプログラムが作れなかったり、モニタアセンブラが常駐しているアドレスへのプログラムが作りにくいなどの問題がありました。DOS版アセンブラではソースからオブジェクトへ「ファイルTOファイル」となっているので、プログラムの大きさ、オブジェクトの置かれるアドレスにほぼ制限がなくなりました。さらに、前述のようにBSAVEファイルも生成できるので、DOS上のプログラムでも、ディスクBASIC上のプログラムでも簡単に作ることができます。

●オプションスイッチ

DOS版アセンブラでは、2種類のスイッチをコマンドライン上に指定することができます。スイッチ1は「モニタアセンブラ」で使用されていたリスト出力のためのもので、スイッチ2はDOS版への改良時点で追加されたものです。

◆スイッチ1

3文字で構成され、1文字めが「プログラムリスト出力」、2文字めが「ラベルリスト出力」、3文字めが「エラーリスト出力」を表します。

各出力に指定できる文字は次の4種類です。

X	……	出力無し
C	……	画面に出力
P	……	プリンタに出力
D	……	ディスクに出力

指定しなければ、すべてが画面に出力されます。〝X〟以外を指定すると処理が遅くなるので、すべてに〝X〟を指定することを勧めます。

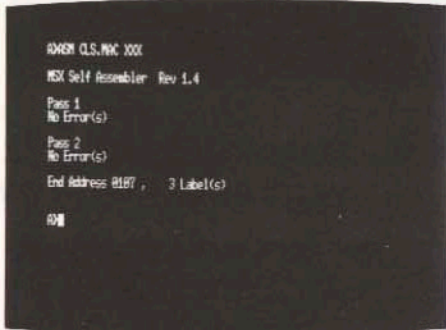
◆スイッチ2

MSX-DOSで一般的に使用されるようなスイッチで〝/〟に続く1文字で構成され、以下のように使用します。

```
A>ASM   TEST.ASM   XCD /N
```

ここで使用できるスイッチと、その意味を以下に示します。

／B	……	BSAVE形式のオブジェクトを出力
／N	……	オブジェクトを出力しない
／Y	……	シンボリックファイルを生成する

各スイッチは、併用することもできますが、〝／B〟と〝／N〟を同時に指定するときなどは矛盾が生じるので、この場合にはどちらかが有効になります(この場合には／Nが有効)。〝／Y〟はシンボリックデバッガ用(S-BUGなど)のファイルです。本書掲載のデバッガでは使用しません。

●オブジェクトファイル

オプションスイッチにて〝／B〟を指定するとBSAVE形式のファイルができますが、このとき拡張子は〝.OBJ〟となります。DOS版アセンブラでは、この他にも2種類の拡張子を持ったファイルを生成することができます。それらをまとめたものが次の表です。

.OBJ	／Bスイッチにより作られる。具体的には、ファイルの先頭に、開始、終了、実行アドレスが入る
.COM	ORG命令を0100hにすると、すぐに外部コマンドとして使えるようにCOMファイルとなる。
.BIN	ORG命令を0100h以外にすると、ただのマシン語ファイルとなる。

●アセンブラの仕様

「モニタアセンブラ」よりの変更は、行番号が不必要になったことと、END命令を必ずつけるようになったことです。他は、同じですが一通り解説しておきます。

◆ソースプログラム

アセンブラのソースプログラムの形式を示します。なお、初めが「行番号 '」となっていると、アセンブラは無視します。

[ラベル：][[ニーモニック][オペランド]][；コメント]

◆文字

ソースプログラムに使用できる文字は文字コードが00H～1FH, 7FH以外のもので、キーボードから入力できて画面にも表示できる文字です。

◆ラベル

ラベルに使用できる文字は、英数字(大文字、小文字の区別はない)、および〝?〟、〝@〟です。ただし、先頭の文字に数字を使用することはできません。ラベルは先頭の6文字が有効で7文字以降は無視されます。ラベルにはレジスタ名、ニーモニック、条件名を使うことはできません。

アドレスを表すラベルは、ラベル名の後ろに〝：〟をつけることで定義されます。定数を表すラベルはEQU擬似命令により定義することができます。

◆ニーモニック

Z80CPUの全命令です(未定義命令は不可)。

◆擬似命令

使用できる擬似命令は次の7個です。DEFB、DEFS、DEFW、DEFM命令において、省略形DB、DS、DW、DMは使用できないので注意して下さい。

・ORG 命令(ORiGin)

ロケーションカウンタ(何番地からオブジェクトを生成するか)を設定します。指定のない場合にはロケーションカウンタが0になります。また、100hにするとCOMファイルが生成されます。

・END 命令

アセンブラに対して、ソースプログラムの終了を指示します。ソースプログラムの最後に必ず入れないとエラーとなります。

・EQU 命令(EQUate)

オペランドの持つラベルを定義します。この命令だけは次の形式で書かれます。ラベルに〝:〞がつかないことに注意して下さい。

```
[ラベル] [[EQU] [式]] [;コメント]
```

・DEFB 命令(DEFine Byte)

メモリに8ビットのデータを設定します。〝,〞で区切ることにより複数個記述することもできます。

・DEFM 命令(DFFine Memory)

メモリに文字列のデータを設定します。文字列は' 'でくくってください。

・DEFW 命令(DEFine Word)

メモリに16ビットのデータを設定します。データは上位8ビットと下位8ビットが逆になってメモリに設定されます。〝,〞で区切ることにより複数個記述することもできます。

・DEFS 命令(DEFine Storage)

オペランドで示されたバイト数の領域を確保します。

◆オペランド

オペランドには、式、レジスタ、これらをカッコで囲んだもの、条件名を書きます。これらの規約を記します。

・式

式は次のような形式をしています

```
[{+|-}] {定数|ラベル|$}
        [{+|-}] {定数|ラベル|$} ……
```

※ [ ] …省略可能であることを示す
{ } …|で仕切られたうちのいずれかを選ぶ

計算は常に16ビットで行われ、オーバーフローは無視されます。結果として1バイトの値が欲しい場合には16ビットのうち下位8ビットが使われることになり、上位8ビットは無視されます。

・定数

定数には数値定数と文字定数があります。

数値定数は0～9で始まる文字列で、10進数と16進数の2種類があります(2進数は使用不可)。データの後に〝D〞をつけるか、もしくは何もつけなければ10進数とみなされます。〝H〞をつければ16進数とみなされます。

16進数の場合、初めの文字がA～Fであったときには、その前に0をつけて下さい(例　A000H ⇒ 0A000H)。

文字定数は1個または2個の文字を引用府(')で囲んだものです。引用府自体は2つ続けることで1つの引用府となります。式の中には3文字以上の文字定数は使用できません。

・$(ロケーションカウンタ)

$は、$をオペランドに記述した命令の1バイト目のロケーションカウンタの値を示します。例えば、「LD HL,$」という命令がD308番地に置かれるとすると、オブジェクトは次のようになります。

アドレス	命令	オブジェクト
D308	LD　HL,　$　⇒	2108D3

・レジスタ名と条件名

レジスタ名および条件名には以下のものが使用できます

レジスタ名	A　B　C　D　E　H　L　I　R AF　BC　DE　HL　IX　IY　SP
条件名	C　NC　Z　NZ　P　M　PE　PO

◆コメント

“;”の後の文字から行の終わりまでの文字列です

◆エラーメッセージ

ソースプログラムに誤りがあると、アセンブラは表3-1で示すエラーメッセージを出力します。

アセンブルを中止するエラー
% Object area full オブジェクトエリアがいっぱいになった
% Label table full ラベルテーブルがいっぱいになった
% Assembler sourse error ソースがアセンブラのソースでない
% Screen not 40×24 text mode SCREEN0でない

アセンブルを中止しないエラー
A　Address Overflow アドレスが0FFFFHをこえた
B　Balance Error カッコの左右の数が合わない 引用符の使い方がおかしい
E　Expression Error 演算子、または式の記述がおかしい
F　Format Error オペランドの数が合わない 該当するニーモニックがないなど
L　Label Error ラベルに予約語を使っている ラベルの記述がおかしい
M　Multiply Defined Label 同じラベルが複数個定義されている
O　Operand Error オペランドの記述がおかしい
P　Phase Error パス1とパス2でラベルの値が異なる
R　Reference Error 相対ジャンプが範囲外 ディスプレイスメントが範囲外
U　Undefined Label 未定義ラベルを参照した
V　Value Error オペランドに記述された値が違っている

表3-1　アセンブルエラー一覧表

3.3 MSX-DOS版テキストエディタ

●テキストエディタの特徴

PED.COM

書式　PED_［<ファイル名>］

アセンブラなどのソースファイルを作成するためのフルスクリーンエディタです。これを使用することにより簡単にテキストファイルを作ることができます。一度に扱える文字は約20K。MSX(1)／MSX2のどちらでも利用可能です。

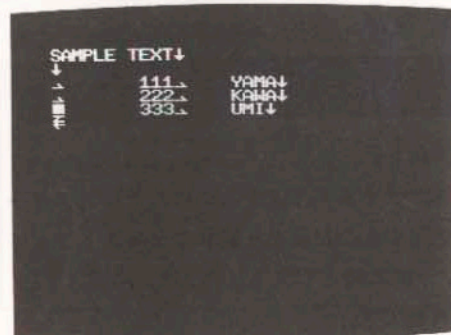
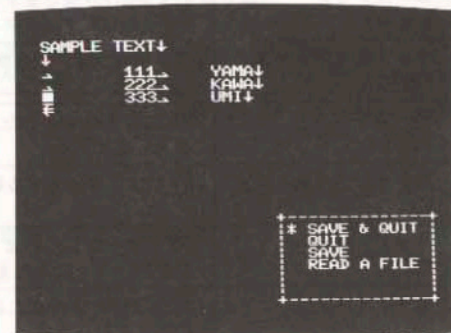
このエディタプログラムは、画面1行の文字数が40文字、または80文字に合わせられて作られています。従って、エディタを起動する前にMODEコマンドを使用して、どちらかの文字数にしておいてください。MSX(1)ならば40文字です。

テキストファイルを更新するときにはファイル名を指定します。新たなファイルを作成したい場合には指定しません。この場合はエディタを抜けるときにファイル名が聞かれます。エディタが起動すると画面がクリアされ、編集ができるようになります。

●エディタの基本的な操作

この状態で、キーボードより文字を入力していきます。試しに、次のようなテキストを入力してみましょう。

```
SAMPLE TEXT ↓
↓
→        1 1 1→    YAMA ↓
→        2 2 2→    KAWA ↓
→        3 3 3→    UMI ↓
```

この中で"→"となっているところはTABキーを押します。また"↓"なっているところはリターンキーです。このエディタではこれらのコードが見えるようになっています。最後にはテキストの終わりを示す"←"があることにも注目してください。

入力できたら、これをセーブしてみましょう。まずESCキーを押してください。画面の右下に四角のウインドウに囲まれたメニューが表示されます。

そこで、カーソルキーの↑↓を使用して、メニュー内の"*"を動かし"SAVE & QUIT"に合わせてリターンします。

ファイル名を指定してない場合にはファイル名を聞いてきますので、答えてください。これでディスクにテキストファイルが作成され、コマンドモードになります。間違いなくできているか確かめるにはTYPEコマンドを使用すればいいでしょう。

●テキストエディタの仕様

本エディタでは、コントロールキーやファンクションキーを用いることにより、さらに柔軟に文章を編集することができます。機能一覧を表3-2に示します。

ファンクションキー	
ロールアップ F1(^U)	行を20行分進めます
ロールダウン F6(^D)	行を20行分戻します
サーチ&リプレース F2(^Q)	文字列の検索、置き換えを行います
ペースト F7(^P)	F8によりカットした文章をカーソル行の前に挿入します
カット&ペースト設定 F3	F8によりカットする範囲を指定します
カット、コピー F8	F3でマークした行より、現在のカーソル行までをバッファに入れます
セットマーク F4	3つのマークのどこにカーソルを合わせるかの指定を行います
ジャンプ F9	F4でセットしたマークのある行へカーソルをジャンプさせます
サーチネクスト F5	F2で設定した文字列を検索します。続けて検索するときに使用します
サーチバック F10	F5がテキストの先方に検索するのに対し、こちらは逆に向かって行います

コントロールキー	
^B	カーソルを文末へ移動します
^C	F8と同じ
^D	F6と同じ
^E	行の終わりまでを削除します
^G	F9と同じ
^H	BS(バックスペース)
^I	TAB(タブ)
^K	カーソル行をカットします。^Yでペーストできます
^L	カーソルをホームポジションに移動します
^M	CR(リターン)
^N	F5と同じ
^O	F10と同じ
^P	F7と同じ
^Q	F2と同じ
^R	オーバーライト/インサート切り替え
^S	F4と同じ
^T	文頭にカーソルを移動します
^U	F1と同じ
^V	カーソルを行の左に移動します
^W	カーソルのある行をカーソルの前行にコピーします
^Y	^Kでカットした行をカーソルの前行にペーストします

その他の機能キー	
ESC	ディスクへの入出力、および終了
TAB	タブコード
RETURN	リターンコード
INS	オーバーライト/インサート切り替え
BS	バックスペース
DEL	デリート
カーソル	カーソル移動

表3-2 エディタの機能キー一覧

◆ ESC

エディタを終えたい、ディスクにロード、セーブしたいという場合に使用します。キーを押したときに表示されるメニューとその意味は以下のとおりです。

* SAVE & QUIT	……	セーブして終了
QUIT	……	セーブせずに終了
SAVE	……	セーブのみ行う
READ A FILE	……	ファイルをロードする

"READ A FILE" を指定するとロードするファイル名が聞かれます。それを入力しリターンするとカーソルのある行の前に指定したファイルがロードされます。

"SAVE & QUIT"、"SAVE" については、コマンド入力時にファイル名を指定しなかったときのみ、ファイル名が聞かれるようになっています。

"QUIT" を指定すると、本当に終えていいのか "SURE？(Y/N)" と聞いてくるので "Y" か "N" で答えてください。

◆カット&ペースト

これは、テキストのブロック単位での削除、移動、コピーを行うための機能です。主に使うキーはF3、F7、F8です。

まずブロックの範囲指定をF3キーで行います。カーソルをブロックの先頭に持っていきF3キーを押します。すると、次のようなメニューが表示されます。

SET MARK	……	マークする
CLEAR MARK	……	マークをクリアする
CLEAR BUFFER	……	バッファをクリアする
MARK=00000	……	マークされた行位置

ここで〝SET MARK〟を選んでリターンすると、その位置が記憶されます。確認はもう1度F3を押すことにより、最後の〝MARK=×××××〟に現れます(この場合リターンしてしまうとマークされた位置が変わってしまうのでESCキーを押して抜けるようにしてください)。

この後、カーソルをブロックの最後に移動し、F8を押します。ここでは、次のようなメニューが表示されます。

CUT	……	ブロックを削除し、バッファに格納する
COPY	……	ブロックをバッファに格納する

〝CUT〟、〝COPY〟は共にブロックをバッファというメモリに記憶するためのものです。違いは、それと同時に元のブロックを削除するか、しないかです。

バッファに記憶したブロックを使いたい場合にはF7キーを使います。まず、カーソルをブロックの移動、コピー先に持っていきます。ここでF7キーを押せばカーソルの前に、先程記憶されたブロックが現れます。

横80ケタでのエディタ画面

これらをまとめると次のようになります。

ブロック削除	①カーソルをブロックの先頭に移動 ②F3キーでブロックの先頭をマーク ③カーソルをブロックの最後に移動 ④F8キーでカット(CUT)する
ブロック移動	①カーソルをブロックの先頭に移動 ②F3キーでブロックの先頭をマーク ③カーソルをブロックの最後に移動 ④F8キーでカット(CUT)する ⑤カーソルを移動先に移動 ⑥F7キーを押してペーストする
ブロックコピー	①カーソルをブロックの先頭に移動 ②F3キーでブロックの先頭をマーク ③カーソルをブロックの最後に移動 ④F8キーでコピー(COPY)する ⑤カーソルを移動先に移動 ⑥F7キーを押してペーストする

◆サーチ&リプレース

テキスト内で、求める文字列を検索したり、置換する機能です。主に使うキーは、F2、F5、F10です。

初めにF2キーを押します。ここでは、次のようなメニューが表示されます。

FIND FORWARD	……	下方に文字列を検索
FIND BACKWARD	……	上方に文字列を検索
REPLACE(ALL)	……	確認無しの置換
REPLACE(STEP)	……	確認有りの置換
F $=*	……	検索文字列
R $=*	……	置換文字列

「検索」の場合には、まず検索文字列を設定します。〝*〟をカーソルキーで〝F $=*〟まで移動させ、リターンし、文字列を入力します。次に、現在のカーソル行よりも下の方に検索したい場合には〝FIND FORWARD〟を、上の方に検索したい場合には〝FIND BACKWARD〟を選択し、リターンします。見つかった場合には、その位置にカーソルが移動し、見つからなかった場合には〝NOT

FOUND.″と表示されます。

検索において、一度のみならず何度か検索したい場合があります。こんなときにはF5またはF10で続行することができます。

「置換」の場合には、″F$＝*″と″R$＝*″を設定する必要があります。ただし、単に検索した文字を削除したい場合には置換文字列を設定する必要はありません。ただし、この場合には確認の意味で″DO YOU MEAN ERACE?(Y／N)″と聞いてきますので″Y″か″N″で答えてください。

それらが終了したら、次は″REPLACE(ALL)″と″REPLACE(STEP)″のどちらかを選びます。″ALL″を選んだ場合は置換操作が自動的に行われ。″STEP″を選んだ場合には文字が検索されたときに″REPLACE?(Y／N)″と聞いてくるようになっています。″Y″なら置換され、″N″ならば置換されません。途中で止めたいときにはESCキーを押してください。

◆エラーメッセージ

INVALID FILE NAME ファイル名のつけかたが間違っていると出力されます。
CHECK DRIVE STATUS ディスクの空き要領がないときに出力されます。
WORNING MEMORY IS INSUFFICIENT! テキスト編集中にメモリがなくなると出力されます。ESCキーで元に戻りますが、テキストを増やす可能性のあるコマンドは受けつけなくなります。
MEMORY IS INSUFFICIENT! TEXT IS SAVED IN PED.PED ファイルを読み込んでいるときにメモリがなくなると出力されます。この場合は、とりあえず読み込んだテキストが″PED.PED″というファイル名でセーブされ、コマンドモードに戻ります。

3.4 MSX-DOS版デバッガ

●デバッガの特徴

BUG.COM

書式　BUG_［<ファイル名>］

マシン語の実行チェックをするためのツールです。1パスアセンブラ、1パス逆アセンブラ、トレーサー、など一通りの機能が備わっおり、充分な威力を発揮してくれます。

デバッガを起動するときに、ファイル名を指定するとそのファイルが0100h番地より読み込まれてデバッガに入ります。そのとき、次のようなメッセージが出力されます。

```
A>BUG TEST.COM
*** MSX DEBUGER ***
0100 0200
-
```

タイトルの後の2つの16進数は、読み込んだファイルの開始アドレスと終了アドレスです。忘れないようにメモしておくことを勧めます。その下の″-″はデバッガのコマンド待ちを示すプロンプトです。

ファイル名を指定しない場合には、開始、終了アドレスは表示されずにコマンド待ちとなります。

●デバッガの仕様

数値の入力についてですが、アドレスおよびデータは16進数で行ってください(″H″は不要)。コマンドの入力は、基本的には大文字でも小文字でもかまいませんが、場合によっては小文字を受け付けないときがあるので気をつけてください。

デバッガで使用できるコマンドは次の14種類です。

A　Assemble

書式　A [<アドレス>]

1パスアセンブラです。入力するとアドレスを表示され、命令が入力できるようになります。命令は1行に1命令づつで、ピリオド(.)でデバッガのコマンド待ちに戻ります。また、命令に誤りがあった場合には、"ERROR" と表示され、やはりコマンド待ちに戻ります。

コマンド入力時にアドレスを指定すると、そのアドレスからアセンブルが始まり、指定しなければ前回入力したアドレスから始まります。

なお、入力はすべて大文字で、値はすべて16進数("H"はつけない)で行ってください。

B　Break point

書式　BP<アドレス>
　　　BL
　　　BC

"BP" はブレイクポイントの設定、"BL" は設定したブレイクポイントの表示、"BC"はブレイクポイントのクリアを行います。"BP"により1度に設定できるブレイクポイントは4個所まで、それ以上設定しようとすると "NO MORE BREAK POINT" とメッセージが表示されます。この場合には "BC" でブレイクポイントをクリアしてください。なお、"BC"コマンドは、使用すると全ブレイクポイントがクリアされますので注意してください。

D　Dump

書式　D [<開始アドレス>] [, <終了アドレス>]

メモリダンプを行います。使用するとメモリの内容を、16進数とキャラクタで表示します。"D" のみだと前回入力したアドレスから1ページ分、"D<開始アドレス>" だと指定したアドレスから1ページ分が表示されます。

F　Fill memory

書式　F<開始アドレス>, <終了アドレス>, <データ>

指定したメモリ間を、指定したデータで埋めます。データは1バイトの値です。

G　Go user program

書式　G [<アドレス>] [, <ブレイクポイント>]

プログラムを実行します。アドレスを指定しないと、Xコマンドで表されるPC値から、指定した場合にはそのアドレスから実行を始めます。実行中にブレイクポイントが設定されている命令を通ると実行が中断されます。

H　Help

書式　H

デバッガにおけるコマンドの簡単な使い方を表示します。

I　set file name

書式　I<ファイル名>

ディスクから読み込んだり書き出したりするファイル名を設定します。Rコマンド、Wコマンドを使用する前に使います。

L　disassemble List

書式　L [<開始アドレス>] [, <終了アドレス>]

メモリ中のプログラムを逆アセンブルして表示します。"L"のみならば前回入力したアドレスから16命令分、"L <開始アドレス>" ならば指定したアドレスから16命令分逆アセンブルして表示します。

Lコマンド実行例
値は16進数ですが"H"はつかないので注意してください

M　Modify memory

書式　M [<アドレス>]

メモリ内容の表示および変更を行います。入力すると、アドレスとその現在の内容を表示しますから、変更する場合には16進数を、変更しない場合にはリターンのみ押します。"."を入力するとデバッガのコマンド待ちに戻ります。

Q　Quit

書式　Q

デバッガを抜けてMSX-DOSのコマンド待ちに戻ります。^Cと入力してもデバッガを抜けることができます。

R　Read user program

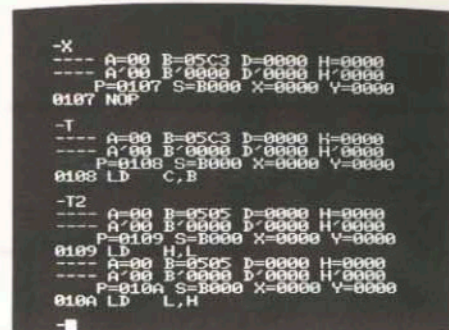
書式　R [<アドレス>]

ディスクからプログラムを読み込みます。アドレスを指定するとそこから、指定しないと0100h番地から読み込まれます。このコマンドを使用する前には、必ずIコマンドでファイル名を指定しておいてください。

T　Trace

書式　T [<カウント>]

マシン語プログラムのトレースを行います。実行はXコマンドで示されるPC値からです。<カウント>を指定しなければ1命令だけ実行されコマンド待ちに戻り、指定した場合には、その数分の命令を自動的に実行します。指定できるカウント数は00～FFまでで、0だと256回トレースされます。どちらも1命令実行するごとにレジスタの内容が表示されます。表示される内容と意味は以下のとおりです。

```
フラグ
↓
S--C   A=00  B=0000  D=0000  H=0000  …表レジスタ群
----   A' 00  B' 0000  D' 0000  H' 0000  …裏レジスタ群
   P=0100  S=AFFF  X=0000  Y=0000
0100  JP    D000  …命令
```


W　Write to disk

書式　W<アドレス1> [, <アドレス2>]

ディスクへプログラムを書き込みます。<アドレス2>を省略すると、0100h番地から<アドレス1>までが、省略しなかったときには<アドレス1>から<アドレス2>が書き込まれます。このコマンドを使用するときにはIコマンドでファイル名を指定しておく必要があります。

X　eXamine

書式　X [<レジスタ名>]

レジスタ内容の表示、および変更を行います。"X"のみならば、Tコマンドと同じ形式でレジスタが表示されます。"X"の後にレジスタ名を指定した場合は、レジスタ値の変更となります。実行すると現在のレジスタ内容が表示され入力待ちとなりますから、設定したい値を入力してください。

```
-XA ⏎          …… Aレジスタを変更するとき
A=00  02 ⏎     …… 設定値を入力
```

レジスタ名は英大文字で入力してください。使用できるレジスタ名は以下のとおりです。

A	Aレジスタ
B	BCレジスタ
D	DEレジスタ
H	HLレジスタ
F	フラグレジスタ

X	IXレジスタ
Y	IYレジスタ
S	SPスタックポインタ
P	PCプログラムカウンタ

※A, B, D, H, Fのレジスタは後に"'"をつけることにより裏レジスタも指定できます。

第6章

MSX-DOS ユーティリティー集

●プログラムの入力の仕方

プログラムの入力は、P 154 掲載の〝COMIN．BAS〟で行います。入力するアドレスは各プログラムの初めを見るか、ダンプリストのアドレスを見てください。また、セーブするファイル名(コマンド名)は、説明の初めの枠で囲まれた名前にしてください。

●多機能モニタ

アドレス 0100H〜230FH　P187

MON．COM

書式　MON_ [-P]

◆概要

MSX2 専用 MSX-DOS 多機能モニタです。スロット切り替え、逆アセンブルなどの機能があります。また、セーブ、ロードをサポートしていますので、COM ファイルのパッチあて(修正)に使用できます。根性があれば COM ファイルも作成できるでしょう。なお〝画面の横文字数は 80 桁にしておいてください。また、ドライブは 3 台以上あると使用できません。プリンタは 80 桁のものを使用してください。81 桁以上のプリンタを使用する場合には〝-P〟オプションを付けて起動してください。

◆起動

起動すると図6-1のような画面が表示されます。

本プログラムは起動されると自分自身を TPA の高位アドレスに転送するようになっています。従って TPA の 100h 番地からモニタの開始番地まで自由に使用できます。モニタの専有するアドレスは画面上の右下にある〝Area〟に表示されます。

図6-1　モニタの画面説明

アスキー・エリア
スロット番号　検索データ　ブロック・アドレス
カーソルのあるアドレス　チェックサム形式

```
Address:0100 Data:0E '.'  Slat #0-0        Find:41 'A'  Block:0000-0000
ADRS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F : 01234567 89ABCDEF : STD8
--------------------------------------------------------------------------------
0100 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 : ........ ........ : 00 00
0110 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 : ........ ........ : 00 00
0120 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 : ........ ........ : 00 00
0130 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 : ........ ........ : 00 00
0140 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 : ........ ........ : 00 00
0150 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 : ........ ........ : 00 00
0160 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 : ........ ........ : 00 00
0170 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 : ........ ........ : 00 00
0180 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 : ........ ........ : 00 00
0190 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 : ........ ........ : 00 00
01A0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 : ........ ........ : 00 00
01B0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 : ........ ........ : 00 00
01C0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 : ........ ........ : 00 00
01D0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 : ........ ........ : 00 00
01E0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 : ........ ........ : 00 00
01F0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 : ........ ........ : 00 00
--------------------------------------------------------------------------------
      AF   SZ-H-PNC   BC   DE   HL   IX   IY    Graphic :    : Area:BB00-D57A
CPU : 0000 (00000000) 0000 0000 0000 0000 0000   window :    : TPA :D606
      v
MEM : 3-2-1-0                                   Copyright(C)1987 by M.GOMI
```

ダンプ・エリア
レジスタ・エリア　TPAの上限
ページ　グラフィック・ウィンドウ　モニタの専有エリア

◆基本操作

a)値の入力

1バイトの値を 16 進数で入力するときには、2 桁入力した後リターンする必要はありません。1桁目で間違えていることに気づいたときは BS キーで戻すことができます。このときキーボードが以下のように設定されているのでテンキーを 16 進入力として使用することができるようになっています(ただし、機種によっては、配列の違いにより入力しにくくなる場合もあります)。

〝.〟 → A	〝,〟 → B	〝+〟 → C
〝-〟 → D	〝*〟 → E	〝/〟 → F

1バイトの値をキャラクタで入力するときには、カーソルが1桁目にあるときにダブルクォーテーション(")を入力することにより次に入力した文字のアスキーコードが設定されます。

2バイトの値は16進数のみで入力します。このときは、16進数4桁を入力した後にリターンキーを押してください。

b)メモリの書き換え

ダンプエリアにカーソルがある状態で、バイト入力を行うと、カーソル位置のアドレスの内容が書き換えられます。正しく書き込めると、カーソルは次のアドレスに移動します。書き換えられなかった場合(ROMであった場合)にはBEEP音がしてカーソルは移動しません。

c)カーソル移動

カーソルキーでカーソルを移動する場合の機能は、次のようになっています。

<table>
<tr><td>①カーソルキーのみ</td><td>→　…　アドレスを1バイト進める
←　…　アドレスを1バイト戻す
↓　…　アドレスを16バイト進める
↑　…　アドレスを16バイト戻す</td></tr>
<tr><td>② SHIFT＋カーソルキー</td><td>→　…　アドレスを4バイト進める
←　…　アドレスを4バイト戻す
↓　…　ロールアップする
↑　…　ロールダウンする</td></tr>
<tr><td>③ CTRL＋カーソルキー</td><td>→　…　検索データを前方にサーチする
←　…　検索データを後方にサーチする
↓　…　アドレスを256バイト進める
↑　…　アドレスを256バイト戻す</td></tr>
</table>

d)各種コマンド(ノーマル)

ダンプエリアにカーソルがある状態でコントロールキーを入力することにより、次のコマンドを使用することができます。

^A(アドレス設定)

表示するアドレスを設定します。現在のアドレスが表示されているところにカーソルが移動するので入力してください。

^C(チェックサム計算方式の変更)

入力するごとにチェックサムの方式が変わっていきます。

MSX　…　MSXマガジン方式(アドレス＋8バイト)
SUM16 …　16バイト単純サム(3桁)
STD16 …　16バイト単純サム(2桁)
STD8　…　8バイト単純サム

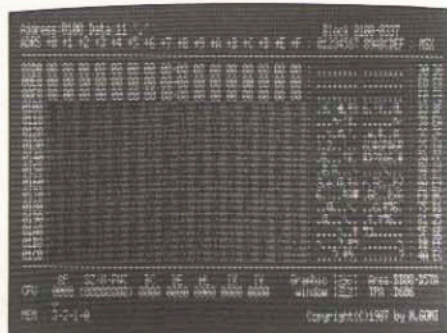
^F(検索データの設定)

画面に"Find"と表示されるので、検索したいデータを1バイト入力します。見つかると画面上でリバース表示されます。解除するときにはもう1度"^F"を押して、リターンします。このコマンドはデータの設定をするだけで、実際の検索はカーソルキーで行います。なお、検索の後述のブロック表示が同時に行われた場合、リバース表示はブロックが優先となるので注意してください。

^G(プログラムの実行)

画面に"Call program"と表示されるので、アドレスを入力します。リターンのみだと、前に設定したアドレスに制御が移ります。キャンセルはESCキーで行ってください。実行はCALL命令で行われます。実行後戻ってくるとレジスタエリアが設定されます。SHIFT＋^Gで実行すると、リターンしたときに入力画面にすぐ戻らないので、画面表示を行っているプログラムのときにはお勧めです。

^L(ファイルのロード)

画面に"Load"と表示されるので、ファイル名を入力します。ファイルは100hにロードされます。サイズチェックをしてないので、巨大なファイルを読み込むときにはTPA以外を壊さないように注意してください。なおSHIFT＋^Lでカーソル位置よりロードすることも可能です。

^R(レジスタ設定)

レジスタエリアの設定を行います。

^W(ファイルのセーブ)

画面に〝Save〞と表示されるので、ファイル名を入力します。セーブされるのは^Lでロードされたのと同じだけのメモリの範囲です。任意のサイズをセーブしたい場合には、後述のブロックセーブを使用してください。

^X(入力モードの切り替え)

カーソルがアスキーエリアに移動し、アスキーキャラクタ入力モードになります。このモードでは入力したキーのコードがそのままメモリに設定されます。カーソルキーのコードもそのまま入力されるので、カーソル移動はできません。^Xで通常のモードに戻ります。

TAB(ヘルプメニュー)

コマンド一覧が表示されます。

^ESC(モニタを抜ける)

モニタを抜けてMSX-DOSに戻ります。

e)各種コマンド(ブロック)

メモリの任意の範囲を対象としたコマンドです。やはりコントロールキーで指定します。

^B(ブロックの開始アドレスの設定)

現在のカーソル位置がブロックの開始として設定されます。設定されるとその位置がリバース表示されます。

^K(ブロックの終了アドレスの設定)

現在のカーソル位置がブロックの末尾として設定されます。設定するとブロック間がリバース表示されます。

^O(ブロックのON／OFF切り替え)

設定したブロックを、有効にするか無効にするか、切り替えます。有効である場合は^M、^P、^S、^Uが使用できます。なお、無効になるとリバース表示されなくなります。

^M(ブロック転送)

ブロックを現在のカーソル位置以降に転送します。転送元と転送先は重なっていてもかまいません。

^P(ブロックプリント)

ブロックをプリンタに打ち出します。実行中画面が変化しますが、問題ありません。ESCキーで中断できます。

^S(ブロックセーブ)

ブロックをセーブします。画面に〝Save〞と表示されるのでファイル名を入力してください。

^U(ブロックフィル)

ブロックを指定した１バイトの値で埋めます。画面に〝Memory Fill〞と表示されるので値を入力します。

f)各種コマンド(プロ)

一歩踏み込んだプロ用のコマンドです。本モニタを最大限に活かすために、これらのコマンドを知っておくといいでしょう。

^D(逆アセンブル)

カーソル位置から16行分逆アセンブルします。数値はすべて16進数で表現されます。また相対アドレスは絶対アドレスに直されます。ここでは、次のキーコマンドが使用できます。

↓　…　次の16行を表示します
↑　…　直前の16行を表示します
^A　…　アドレスを設定し直します
^P　…　ハードコピーを行います

本モードを抜けるときはESC、スペース、リターンを押します。ESCかスペースにより「本モードに入ったときのアドレス」、リターンにより「現在逆アセンブルしたアドレス」、に戻ることができます。

^T(スロット切り替え)

デフォルトのスロット(MSX-DOSのスロット)と^Yで指定したスロットを切り替えます。

^Y(スロット番号の指定)

画面に〝Slot〟と表示されるので、初めに基本スロット番号(0～3)を入力します。その後、拡張スロット番号を指定する場合はそれを入力(0～3)し、そうでない場合はリターンのみ押します。正しく入力されると、その場でスロットが切り替わります。なお、スロットが切り替わっているときは、検索、実行、セーブ、ロード、プリントを除くブロックコマンドは無効となります。また、スロット変更中はスクロール速度が遅くなります。

^Vn(アドレスの記憶)

現在のアドレスを記憶します。nは1～3の値で、3か所の記憶が可能となっています。

^Qn(アドレスの移動)

^Vnで記憶したアドレスに移動します。nは1～3の他に「B…ブロックの先頭」、「K…ブロックの末尾」、「Q…現在のアドレスを^V3として記憶してから^V3で記憶したエリアに移動する」が使用できます。

^QP(ハードコピー)

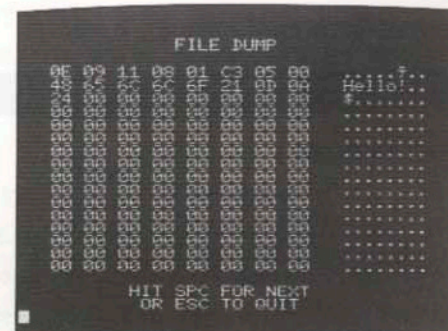
画面のハードコピーをプリンタに打ち出します。

●ファイルダンプ

アドレス0100H～027FH　P196

FDUMP.COM

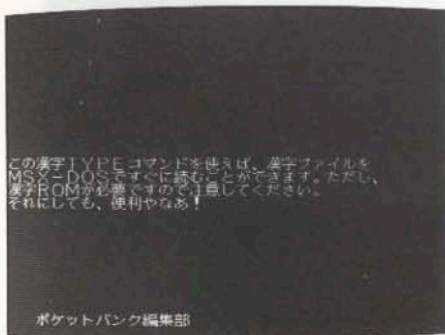
書式　FDUMP_<ファイル名>

ファイルの内容を画面に16進数で表示するためのコマンドです。1画面ごとに表示は止まりますので任意のキーを押してください。ファイルを最後まで表示すると終わります。

●漢字 TYPE

アドレス 0100H～0BD7H　P197

KTYPE . COM

書式　KTYPE_<ファイル名>

シフト JIS で書かれた漢字ファイルを画面に表示します。このプログラムを使うときは漢字 ROM が装備されてないと使えません。MSX2 専用となっています。

●ディスクダンプ

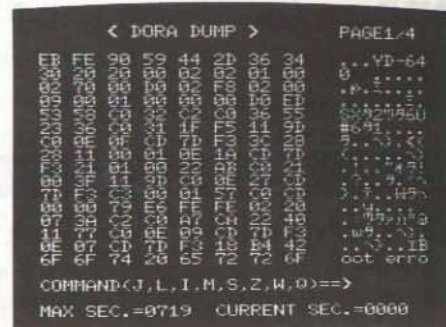
アドレス 0100H～06FFH　P200

DKDUMP . COM

書式　DKDUMP

ディスクの内容をセクタごとに表示するためのユーティリティーです。実行すると以下のような画面が表示されます。

ドライブ番号を聞いてくるので〝A〞とか〝B〞と指定してください。リターンすると、初めに論理セクタ 0 が読み込まれ、その内容が 16 進数とキャラクタで表示されます。1つのセクタは 512 バイトですが、1画面に入らないので、4つのページに分けてあります。

画面の〝MAX SEC.〞は論理セクタの上限値、〝CURRENT SEC.〞は現在表示されている論理セクタ、を表しています。ここでは次のコマンドを使用することができます。

J、←	…	セクタ間の移動(若番方向へ)
L、→	…	セクタ間の移動(老番方向へ)
I、↑	…	ページ間の移行(若番方向へ)
M、↓	…	ページ間の移行(老番方向へ)
S	…	セクタ番号の指定
Z	…	データの変更
W	…	データの書き出し
Q	…	終わり

Z コマンドはディスクのデータを変更するコマンドです。画面上にカーソルが現れるので変更したいところに動かして上書きしてください。ただし、この時点ではディスクに書き込まれないので ESC キーを押してコマンド入力に戻り、W コマンドを実行してください。

●簡単ファイルコピーその1、その2

アドレス0100H〜127FH　P202
アドレス0100H〜1407H　P207

FCOPY.COM／FCOPY2.COM

書式　FCOPY／FCOPY2　[<ファイル名>]

コピーしたいファイルを選択して、AドライブからBドライブに一気にコピーするユーティリティーです。「その1」はMSX(1)／2兼用、「その2」はMSX2専用です。コマンドを入力すると次のような画面が表示されます。「その1」の方はファイル名が2列に表示され、「その2」の方は5列に表示されます。コマンド入力時、「その2」の方はファイル名が指定でき、〝FCOPY2　*．COM〞などとすることにより一定の名前だけあらかじめ選択することができます。

FCOPYのファイル名選択

FCOPY2のファイル名選択

選択にはスペースキーを使います。押すと「その1」の場合ファイル名の横に〝*〞マークが、「その2」の場合ファイル名が反転表示され、コピーするファイルであることが設定されます。もし間違えてつけてしまった場合には、もう1度スペースキーを押せば消えます。またMキーを使って選択することもできます。この場合は1度目にMキーを押したところから2度目に押したところまでの連続した部分が選択されます。また、「その1」で、ファイルが画面に入りきらない場合は、スクロールするようになっています。なお、選択中やめる場合はESCキーを押してください。

●CP／M→MSX-DOSファイルコンバート

CP／MとMSX-DOSのディスクの管理の仕方は、根本的に異なります。そのため、CP／M上のファイルをMSX-DOS上に転送しようとすると、非常に困難を究めます。そこで、ここではCP／MマシンとMSXとをRS-232Cでつないで、それを通して転送することを試みました。

転送作業をするのには、RS-232CつきのCP／Mマシンと、RS-232CつきのMSXです。また、CP／Mの方には転送用プログラムを作成するためM80などのアセンブラが必要です。ちなみに、ここで使用したシステムは「PC-9801+*PLUS80」と「RS-232CつきMSX」です。

なお、説明を読んで意味がわからない人は行わないでください。

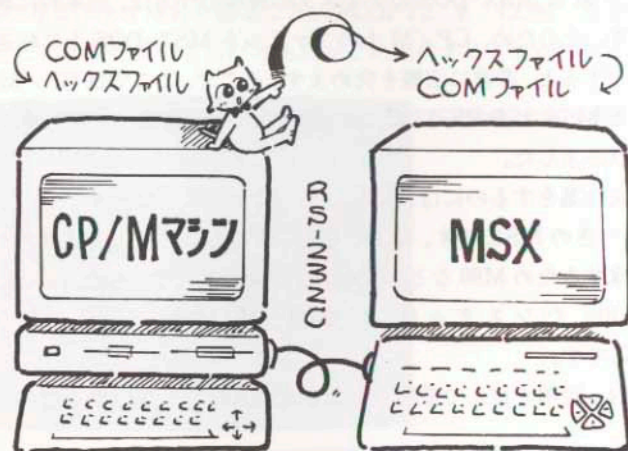
◆転送する手順

RS-232Cを通しての転送は、アスキーファイルに限ることにします。そのため、まず目的のプログラムファイル(例えばCOMファイル)をアスキーファイルに直す作業から始めます。このとき採用するのは「ヘックスファイル形式」で、ファイルの中味を全部16進数の文字列に変換したものです。これは、ファイルの中の1バイトを見たときに、例えばそれが1Ahであったなら、これを文字の〝1〞と〝A〞に直して書き出したファイルをいいます。

*カノープス電子(株)発売のPC-9801用Z80ボードです。

"BINTOHEX．MAC"（P 212）がそのためのプログラムです。これは、CP／M上で実行し、目的のファイルをヘックス形式に変換します。

後の手順は簡単で、このヘックスファイルをRS-232Cを通してMSXに送ってやり、次にMSXの方でこのヘックスファイルをもとのファイル形式に直してやればいいわけです。

CP／MからRS-232Cを通して送り出すためのプログラムが"CTRANS．MAC"（P 218）で、MSXの方で受け取るためのプログラムが"MTRANS．BAS"（P 221）です。また、それを元のファイル形式にに戻すためのプログラムが"HEXTOBIN．BAS"（P 222）です。

◆使用方法

例として、CP／MからCLS．COMという名前のファイルをMSXに転送してみることにします。

①CP／Mシステムの方で何らかのエディタを使って、"BINTOHEX．MAC"、"CTRANS．MAC"を入れ、それをアセンブルして2つのCOMファイル("BINTOHEX．COM"、"CTRANS．COM")を作ります。

②MSXのディスクBASICで、"MTRANS．BAS"、"HEXTOBIN．BAS"を入力しておきます。

③MSXとCP／Mマシンを、RS-232C用クロスケーブルを使ってつなぎます。

④CP／Mシステムの方で、目的のファイルを"BINTOHEX．COM"使ってヘックスファイルに変換します。

```
A>BINTOHEX　CLS.COM
```

これで"CLS．HEX"というファイルが作られます。

④MSXのディスクBASICで"MTRANS．BAS"を実行し、続いてCP／Mの方で、コマンドラインから次のように入力します。

```
A>CTRANS　CLS.HEX
```

するとCP／MからMSXに"CLS．HEX"が転送され、MSX側はそれを"GOTCPM．HEX"とい名前のファイルに書き出します。

⑤転送が終ったら、MSXのディスクBASICで"HEXTOBIN．BAS"を実行します。そうすると"GOTCPM．COM"という名前のファイルが作られます。後は適当なファイル名にこれをリネーム(REN)すれば終わりです。

CP／Mの方で元々アスキーファイルだったものは、いちいちヘックスファイルに直さなくても、"CTRANS.COM"を使って転送できます。この場合はMSX側で"HEXTOBIN．BAS"にかける必要もないため"GOTCPM．HEX"をそのままリネームすればいいでしょう。

なお、"CTRANS．COM"を使って転送する前には、通信方式を設定しなくてはいけません。MSXの方は"TRANS．BAS"の中で次頁のように設定しているので、CP／Mの方もそれに合わせて設定するといいでしょう。自分でCP／MとRS-232Cのマニュアルを読んで適切に設定してください。

項目	設定
ボーレート	…1200
キャラクタ	…８ビット
パリティ	…なし
ストップビット	…１
XON／XOFF	…なし
ＣＳ	…なし
オートLF	…なし
SI／SO	…なし

なお、転送されたプログラムは正常に動かない場合もあります。原因は、元からMSX-DOSでは動かないプログラムか、あるいは転送中にノイズを拾ってしまったかのどちらかだと思われます。何度か手順等を確認し、やり直してみて、ダメだったときには素直にあきらめるのが賢明だと思われます。

最後に書いておきますが、CP／M上の市販のプログラムは、コピーしてはいけないことになっています。気をつけてください。

参考文献

MSXテクニカルハンドブック（アスキー）
MSX-DOS-TOOLSマニュアル（アスキー）
MSXポケットバンクシリーズ（アスキー）
MSXマガジン（アスキー）

プログラム・スタッフ

ASM	.COM	大貫広幸／長井健一
PED	.COM	張　東一
BUG	.COM	清水　真佐志
MON	.COM	五味　守
FDUMP	.COM	張　東一
KTYPE	.COM	長井健一
DKDUMP	.COM	張　東一
FCOPY	.COM	長井健一
FCOPY2	.COM	長井健一
その他		B I T S

表紙イラスト

ケロヨン村田

本文イラスト

茶目高橋

本文写真

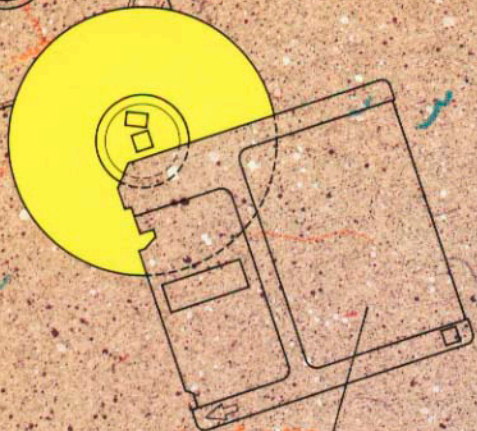
山田裕司

MSX-DOS
スーパーハンドブック

1987年11月9日　初版発行／1988年4月1日　2版発行
定価1,200円

著　者　BITS
発行者　塚本慶一郎
発行所　**株式会社アスキー**
〒107 東京都港区南青山6-11-1 スリーエフ南青山ビル
振　替　東京4-161144
TEL (03)486-7111 (大代表)
出版営業部TEL (03)486-1977 (ダイヤルイン)

編集担当　秋山耕一
CTS　株式会社福田工芸
印刷　株式会社加藤文明社

ISBN4-87148-045-3 C3055 ¥1200E

■COMファイル入力用ツール

■COMファイルプリントアウトツール

■DOS版テキストエディタ

■DOS版アセンブラ

■DOS版デバッガ

■多機能モニタ

■ファイルダンプ

■漢字TYPE

■ディスクダンプ

■簡単ファイルコピーその1

■簡単ファイルコピーその2

■CP/M用ファイルTOヘックス

■CP/M用送信プログラム

■MSX用受信プログラム

■MSX用ヘックスTOファイル

ISBN4-87148-045-3 C3055 ¥1200E

定価1200円